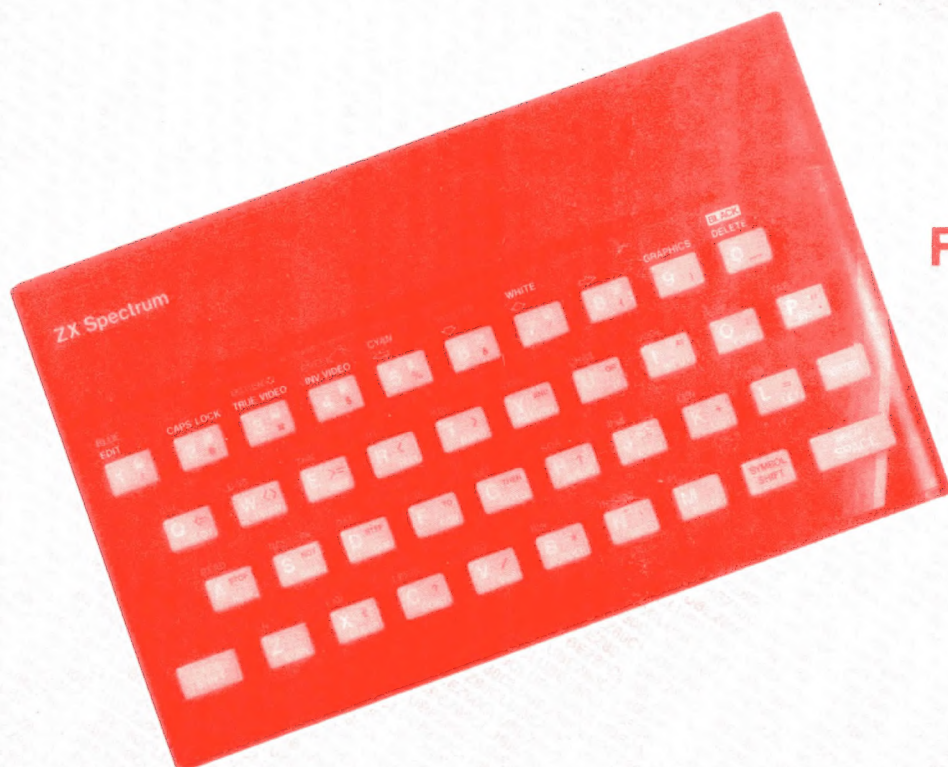


CLUBE

Z

80



Fevereiro/84

N.º 17

NESTE NÚMERO

INTRODUÇÃO À LINGUAGEM MÁQUINA	1
ESPAÇO SPECTRUM	4

Programas Spectrum/ZX81/Newbrain

Piramide	9
Colony Invaders	10
Batalha Naval	11
Frequências	13
Casa Assombrada	13
Perigo no Espaço	15
Submarinos	16
3 D-O'S-X'S	17
COMO CONTROLAR O GRAVADOR COM O SPECTRUM	19
NOVOS PROGRAMAS E LIVROS	21

No Interior:

Cupão de Inscrição

Edição: Clube Z80

Fotocomposição: Fotomecânica Mabreu/Porto

Impressão: Gráfica Firmeza/Porto

Tiragem: 500 exemplares, Fevereiro 1984

INTRODUÇÃO À LINGUAGEM MÁQUINA

ZX81

Autor: FERNANDO PRECES
Sacavém

(Continuação do número anterior)

A segunda rotina em Código Máquina, cujo diagrama-bloco já apresentámos, inicia as variáveis ED, CB e X. Há vários processos de tratar as variáveis na feitura de uma rotina em CM; porém o processo mais simples é instalar cada variável numa localização da RAM e etiquetá-las devidamente. Podemos elaborar a listagem inicial desta parte da rotina em formato Assembler como se segue:

ETIQUETAS	MNEMÓNICAS	COMENTÁRIOS
ED	—	} Endereços destinados às variáveis
CB	—	
X	—	
Y	—	
	LDA,+1	
	LD(ED),A	para ED = 1
	LD(CB),A	para CB = 1
	INC A	
	INC A	
	LD(X),A	para X=3

O passo seguinte é tratarmos a operação PLOT X,Y. É necessário:

- 1) Carregar o registo BC com as coordenadas X e Y, o valor de X no registo C e Y no B.
- 2) Comutar o sistema de variáveis (T-ADDR) para PLOT.
- 3) Introduzir uma instrução CALL para a rotina de comando PLOT.

As rotinas de comando PLOT e UNPLOT (na ROM) exigem 2 testes. O primeiro verifica se os parâmetros X e Y estão escritos na escala correcta. O segundo se a operação é PLOT ou UNPLOT.

O primeiro teste é em tudo semelhante ao que é efectuado na rotina PRINT AT, já nossa conhecida. O segundo é um pouco mais complicado, mas vamos segui-lo:

As instruções PLOT e UNPLOT desencadeiam o trabalho de uma rotina que se situa na ROM, endereço 2994. Esta começa por memorizar o endereço da instrução UNPLOT que se encontra na *tabela das instruções de comando* (também na ROM), que é introduzido no registo DE, e coloca no registo A o conteúdo de (T-ADDR) do sistema de variáveis (endereços 16432/3).

A comparação do baixo byte do registo DE com o conteúdo de A resulta num salto.

Se o resultado for negativo significa que a instrução de comando é PLOT; se for positivo, na continuidade da sequência, significa que a instrução é UNPLOT. Quando se trabalha em linguagem máquina, por ve-

zes podemos interditar o programa monitor (que foi concebido para desempenhar as funções de control em programação BASIC) de actuar, poupando assim tempo de vai e vem.

Mas nem sempre isto é possível, por haver instruções muito vigiadas que nos impedem ("by pass" — expressão inglesa que significa *passar por fora*). Nesta rotina somos forçados a cumprir o especificado pois se não o fizemos o ZX não compreende a instrução. Assim, se queremos uma instrução PLOT, teremos de:

- 1.º — Colocar no registo BC os parâmetros X e Y.
- 2.º — Colocar no registo A, o valor 9B (decimal 155).
- 3.º — Introduzir na variável do sistema (T — ADDR), o conteúdo de A.
- 4.º — Fazer um CALL para a rotina de comandos PLOT e UNPLOT.

O passo seguinte da nossa segunda rotina será o de pesquisar se as teclas (R) ou (BREAK) foram premidas.

NOTA: Aqui encontrará o leitor uma primeira solução que lhe permite equacionar o problema de como mover a nave da terra, no jogo 1, com uma rotina em CM.

Uma instrução CALL para a rotina de *pesquisa do teclado* (na ROM) faz o retorno do valor da tecla para dentro do registo HL. Esse valor será depois testado para ver se é (R) ou (BREAK), ou se nenhuma tecla. As linhas seguintes em pseudo-BASIC, mostram-nos uma das várias soluções, talvez a mais simples, para essa operação.

NOTA: No interior de HL podem aparecer os seguintes valores:

- 1 — Para nenhuma tecla premida (—1)
- 2 — Para (R) premido (61435)
- 3 — Para (BREAK) premido (64895)

```

10 LET HL = (um dos 3 valores)
20 LET DE = HL troca dos registos
30 LET HL = 61435 valor de (R)
40 LET HL = HL — DE Testa (R)
50 IF HL = 0 THEN RETURN
60 LET HL = 64895 valor de (BREAK)
70 LET HL = HL — DE Testa (BREAK)
80 IF HL = 0 THEN RETURN
   etc... se o valor de HL não for nem (R) nem (BREAK)

```


Eis a listagem assembler correspondente:

ETIQUETAS	MNEMÓNICAS	COMENTÁRIOS
Teste do teclado	CALL (Teclado)	ROM — Endereço 699
	EX DE, HL	Troca conteúdo registos
	LD HL, + 61435	Código Tecla (R)
	AND A	(Ver nota)
	SBC HL, DE	Subtrai HL — DE
	RET Z	Retorno se (R) premido
	LD HL, + 64895	Código Tecla (BREAK)
	AND A	
	SBC HL, DE	
	RET Z	Retorno se (BREAK) premido

NOTA: (instrução AND A) — Tal como outras instruções para operações lógicas, será devidamente abordada noutro capítulo deste curso. No entanto o seu aparecimento nesta rotina necessita de uma chamada de atenção e dum leve esclarecimento, para não deixar o leitor a braços com o porquê da sua utilização.

O registo F, também de 8 bits, não é utilizado para armazenar dados, como os restantes.

Individualmente os seus bits são usados como indicadores de estado ou "flags", servindo de complemento a muitas instruções do Z 80.

A instrução de subtracção SBC, utilizada nesta rotina, é afectada pelo estado de um desses indicadores. Há assim necessidade de colocar esse indicador a 0 (RESET) antes de iniciar a operação.

Existem vários processos para *limpar* (colocar a 0) o tal indicador do registo F, sendo a instrução AND A, a mais utilizada.

Mas continuemos...

As linhas em pseudo-BASIC a seguir esclarecem-nos como fazer o teste de (X):

10 LET ED = 1	Define ED
20 LET X = 3	Define X
30 LET A = X	LDA, (X)
40 IF A = 3 THEN GOTO 60	Testa muro esquerdo
50 IF A <> 61 THEN GOTO 80	Teste muro direito
60 LET A = ED	LDA, (ED)
70 LET ED = - ED	a inversão
etc.	

Podemos fazer igual raciocínio para o teste do (Y), onde a diferença está somente nos valores a definir. Assim a primeira listagem em Assembler para este conjunto, será:

ETIQUETAS	MNEMÓNICAS
Teste de (X)	LDA, (X)
	CP, + 3
	JRZ, (Inversão de ED)
	CP, + 61
	JRNZ, (Teste de Y)
	LDA, (ED)
	CPL (Ver nota)
	INCA (Ver nota)
	LD (ED), A
Inversão de ED	

Teste de (Y)

Inversão de CB

LDA, (Y)
CP, + 6
JRZ, (Inversão de CB)
CP, + 41
JRNZ, (UNPLOT)
LDA, (CB)
CPL } (Ver nota)
INCA }
LD (CB), A

NOTA: Neste conjunto de Assembler, acima representado, aparecem 2 vezes as instruções CPL e INC A para fazer a inversão de ED e CB. Bastante há a dizer sobre a utilização conjunta destas duas instruções, mas por agora podemos informar o leitor que efectuem uma operação dentro do registo A, mudando o sinal do seu conteúdo, de mais para menos. Existem duas instruções nas mnemónicas do Z 80 que podem causar confusão ao leitor. A primeira, código 47 (CPL), inverte o estado de todos os bits do registo A, e chama-se o complemento de A. A segunda (CP L), código 189, é uma instrução que compara o conteúdo de A com o conteúdo do registo L.

O caso que citámos coloca o registo A, (CPL) código 47, a funcionar não em binário absoluto, como é normal, mas em 2.º complemento aritmético, quando antecede uma instrução INCA.

Sempre que trabalhamos com o 2.º complemento aritmético, o ZX sabe que o estamos a utilizar.

Estas, como outras instruções complexas do Z 80, terão de ser amplamente debatidas, se quisermos saber programas em código máquina. No segundo e terceiro: capítulos deste texto, encontrará o leitor o aprofundamento teórico e prático de cada uma dessas instruções.

Continuando com o jogo...

A operação UNPLOT é idêntica à operação PLOT; porém a constante a introduzir em T-ADDR, tem de ser alterada.

ETIQUETA	MNEMÓNICAS
UNPLOT	LDBC, (X e Y)
	LD A, + 160
	LD (T — ADDR), A
	CALL (Rotina de comando PLOT E UNPLOT)

Por último, entramos no desenvolvimento do conjunto de instruções que vão actualizar durante o jogo os valores de X e Y.

Também esta parte da rotina pode ser escrita por vários processos. Foi escolhido um dos mais utilizados.

ETIQUETAS	MNEMÓNICAS	COMENTÁRIOS
Valor de X	LD HL, + ED	Endereço de ED
	LD BC, + X	Endereço de X
	LD A, (BC)	Coloca em A, o valor de X
	SUB (HL)	$X = X - ED$
Valor de Y	LD (BC), A	Restaura X
	DEC HL	Endereço de CB
	INC BC	Endereço de Y
	LD A, (BC)	Coloca em A o valor de Y
	SUB (HL)	$Y = Y - CB$
	LD (BC), A	Restaura Y
	JR (PLOT)	

Estamos agora em condições de fazer uma listagem completa das instruções que compõem a 2.^a rotina.

ETIQUETAS	MNEMÓNICAS	COMENTÁRIOS
CB	—	Localização das 4 variáveis
ED	—	
X	—	
Y	—	
Valores	LD A, + 1	Atribui o valor 1 às variáveis CB e ED
	LD (CB), A	
	LD (ED), A	
	INC A	
PLOT	INC A	Atribui o valor 3 à variável X
	LD (X), A	
	LD BC, (X)	
	LD A, + 155	
	LD (T - ADDR)	Valor de tabela a atribuir para PLOT que é colocado em T-ADDR
	CALL PLOT	Rotina de comando PLOT na ROM
KEY TEST (teste teclado)	CALL TECLADO	Rotina de comando na ROM
	EX DE, HL	Troca de conteúdos
	LD HL, + 61435	Valor da tecla "R"
	AND A	
	SBC HL, DE	
	RET Z	
	LD HL, + 64895	Valor da tecla "BREAK"
	AND A	
TESTE DE X	SBC HL, DE	
	RET Z	
	LDA, (X)	Testa o lado esquerdo
	CP, + 2	Se for 0 salta p/ a inversão
ED-Reverse	JR Z, (ED-REV)	Testa o lado direito
	CP, + 61	Salta se < > de 0
	JR NZ, (Teste de Y)	
TESTE DE Y	LDA, (ED)	Inversão direcção da Bola
	CPL	
	INC A	
	LD (ED), A	
CB-Reverse	LD A, (Y)	Inversão direcção da Bola
	CP, + 6	
	JR Z, (CB-REV)	
	CP, + 41	
	JR NZ, (UNPLOT)	Salta se < > de 0
	LD A, (CB)	
	CPL	
	INC A	
	LD (CB), A	

UNPLOT	LD BC, (X)	Valor de tabela a atribuir para UNPLOT
	LD A, + 160	
	LD (T - ADDR), A	Rotina comando PLOT e UNPLOT para apagar o rasto da Bola.
	CALL PLOT	
Novo valor X	LD HL, + (ED)	Valor de X actualizado
	CD BC, + (X)	
	LD A, (BC)	
	SUB (HL)	
Novo valor Y	LD (BC), A	Sobe um endereço p/ (CB)
	DEC HL	
	INC BC	
	LDA, (BC)	
	SUB (HL)	Desce um endereço p/ (Y)
	LD (BC), A	
	JR PLOT	Valor de Y actualizado

Esta rotina em CM executa o trabalho das linhas Basic (140 a 160) e (180 a 250).

ESCREVA:

15 REM XXX (Reserve 113 caracteres)

Introduza este código:

16574	0,	0,	0,	0,	62,	1,	50,	60,	64
16583	50,	191,	64,	60,	60,	50,	192,	64	
16591	237,	75,	192,	64,	62,	155,	50,	48	
16599	64,	205,	178,	11,	205,	187,	2,	235	
16607	33,	251,	239,	167,	237,	82,	200,	33	
16615	127,	253,	167,	237,	82,	200,	58,	192	
16623	64,	254,	2,	40,	4,	254,	61,	32	
16631	8,	58,	191,	64,	47,	60,	50,	191	
16639	64,	58,	193,	64,	254,	6,	40,	4	
16647	254,	41,	32,	8,	58,	190,	64,	47	
16655	60,	50,	190,	64,	237,	75,	192,	64	
16663	62,	160,	50,	48,	64,	205,	178,	11	
16671	33,	191,	64,	1,	192,	64,	10,	150	
16679	2,	43,	3,	10,	150,	2,	24,	159	

TERMINA EM 16687

Recapitulando o BASIC:

```

10 REM XXX (53 caracteres)
15 REM XXX (113 caracteres)
20 LET K =USR 16533
30 POKE 16577, INT (RND * 33 + 8)
40 LET K =USR 16578
50 RUN

```

E pronto, a primeira parte do jogo está pronta a funcionar.

O leitor a partir daqui pode introduzir muito mais coisas.

Por exemplo:

- 1 — Introduzir Alvos de Impacto
- 2 — Introduzir Raquetes
- 3 — Introduzir Contador para Pontos

E muitas mais coisas podem ser imaginadas para este jogo.

Final do primeiro Capítulo

(Continua)

TROCO QUALQUER TIPO DE PROGRAMAS PARA O ZX SPECTRUM.

POSSUO CERCA DE 25 PROGRAMAS, DESDE UTILITÁRIOS (COMPILADOR BASIC, VU-3D, ETC.) ATÉ DIVERSOS JOGOS (PENETRADOR, HORACE SKIING, ETC.).

ANTÓNIO MANUEL DE BASTOS PEREIRA

CABÊÇO — PESSEGUIERO DO VOUGA

3740 SEVER DO VOUGA

ESPAÇO SPECTRUM

Autor: FERNANDO D'ALMEIDA PRECES

CAPÍTULO I

PARTE IV

Parte 4 — Pequenos programas monitores

Muitos programas deste tipo, cuja utilidade pode à primeira vista parecer de pouco interesse, vem mais cedo ou mais tarde a tornar-se numa peça importante do nosso arquivo, desde que tenhamos um processo prático e rápido de os transportarmos para o computador, na ocasião em que deles precisamos.

Portanto o problema situa-se mais ao nível da organização dum arquivo e processos de localização dum pequeno programa, do que falta de tempo para o escrever no computador.

Mesmo que o leitor possua poucos programas gravados deve, por método, elaborar um ficheiro que os organize.

De início podemos fazê-lo com um simples caderno de folhas soltas aonde, como primeira página, devemos criar um índice que agrupe as cassetes por especialidades.

As páginas seguintes agruparão programas de uma dada cassette e, nesta conformidade, teremos uma fita "n" aonde vamos gravar todos os pequenos programas monitores.

Cada um desses programas, chamados por um número, será elaborado a partir da linha Basic 9900 e conterá o mesmo número de linhas de forma a que possamos, para um certo trabalho, chamar em continuidade "n" programas e no final somente ter de eliminar o grupo de linhas Basic correspondentes ao último monitor introduzido.

O tempo de gravação de cada um desses programas é bastante curto para permitir uma introdução rápida no computador. Um conta voltas incorporado no gravador acelera o processo, desde que nas páginas do

ficheiro se encontre mencionado o número de rotações aonde começa o programa pretendido.

Os programas monitores que seguidamente vamos introduzir neste texto, encontram-se baseados nas normas especificadas.

PROGRAMA "1" — Formação duma REM com o número de linha Ø, para reserva de "n" bytes.

Apesar de algumas desvantagens que à frente vamos enumerar, este processo para a introdução de código máquina é ainda utilizado por alguns programadores. Pessoalmente, utilizo a REM Ø para introdução de texto (instruções e legendas), a visualizar em determinada sequência dum programa, apenas enquanto este se encontra na fase de programação.

Normalmente nessa fase, o texto de apresentação e as legendas podem sofrer alterações que afectam a sua extensão. Então, se este estiver já enquadrado entre rotinas de C.M., poderá ser difícil arranjar espaço para a sua ampliação.

Uma vez terminado o programa, podemos *transportar* o conteúdo da REM, para qualquer ponto da RAM, de forma a uni-lo às rotinas em C.M., já concluídas.

O nosso primeiro monitor — o construtor de REMS — está escrito em Basic, e as expressões matemáticas nele contidas não se encontram simplificadas, (isto é, reduzidas à expressão mais simples) de forma a auxiliar o leitor menos treinado a seguir as suas combinações, desde que conheça os rudimentos da formação duma linha em Basic, neste computador. Se tal não acontecer, terá de ser paciente e aguardar, pois lá chegaremos.

O programa ao correr, começa por inquirir quantos *bytes quer reservar*. Se o número indicado for *inferior a 354* (o que acontecerá na maioria dos casos), não são introduzidas instruções de manobra, e o progra-

ma segue a sequência normal pedindo no final para apagar as REMS não utilizadas.

Se esse número for igual ou superior a 354, terá de seguir as instruções que o programa colocará no écran, após os primeiros cálculos.

```

1 REM 12345678901234567890123
4567890
2 REM 12345678901234567890123
4567890
3 REM 12345678901234567890123
4567890
4 REM 12345678901234567890123
4567890
5 REM 12345678901234567890123
4567890
6 REM 12345678901234567890123
4567890
7 REM 12345678901234567890123
4567890
8 REM 12345678901234567890123
4567890
9 REM 12345678901234567890123
4567890
10 REM 12345678901234567890123
4567890
9900 BORDER 5: PAPER 5: INK 0: C
LS
9905 PRINT AT 2,0;" APROVEITAMEN
TO DA 1a. REM PARA" TAB 3;"ARMA
ZENAR DADOS."
9910 LET i=23755
9915 LET t=n*36+(i-1): LET b=n*3
6-4
9920 LET c=INT (b/256): LET d=((
b/256)-c)*256
9925 LET k=t-(i+5)
9930 POKE i+2,d: POKE i+3,c
9940 FOR n=i+5 TO t-1: POKE n,45
: NEXT n: POKE 23756,0
9945 PRINT "," "Estão reservados
na 1a.REM, ",";k;" Bytes."
9950 PAUSE 200: CLS : LIST : STO
P
9955 BORDER 5: PAPER 5: INK 7: C
LS
9960 PRINT AT 2,0;" APROVEITAMEN
TO DA 1a. REM PARA" TAB 3;"ARMA
ZENAR DADOS.": PRINT AT 9,3;"IND
IQUE QUANTOS BYTES QUER" TAB 11
;"RESERVAR?": INPUT K
9965 PRINT " INK 1;TAB 3;"=> In
dicados ";k;" Bytes"
9970 LET n=1+INT ((K+6)/36): IF
n<=10 THEN GO TO 9990
9975 INPUT "Prima uma tecla para
instruções. ";l$: CLS
9985 PRINT AT 2,11;"INSTRUÇÕES")
"" Chame a linha 10, mude o núm
ero de linha para 11 e repita a
operação ";n-10;" vezes."" " Qua
ndo terminar a sua intervenção
tera ";n;" REMS iguais."" " Depo
is GO TO 9900."" : STOP
9990 PRINT " INK 0;" Prima uma
tecla para accionar o programa.""
"" Depois apague as REMS não uti
lizadas.": INPUT l$: CLS : GO TO
9900
9999 SAVE "1" LINE 9955

```

PROGRAMA "2" — Introduzidor de texto e legendas numa REM 0.

Este programa deve ser chamado com a instrução MERGE "2" para apagar todas as linhas do programa anterior com excepção da REM 0.

Tem incorporadas as instruções necessárias para a sua manipulação, mas será útil chamar a atenção do leitor para o seguinte:

- As palavras que vamos introduzir na REM serão mais tarde para aparecer no écran, ao longo dum programa, e têm portanto de estar devidamente *arranjadas* de forma a preencher blocos exactos de 32 caracteres, tal como fazemos quando empregamos uma instrução PRINT.
- No final do texto ou legendas, teremos de imprimir o símbolo "#" (Código 35).
- Se por lapso a extensão do texto ultrapassar o número de bytes reservado, o programa encrava protegendo o que está feito. Neste caso, e também para efectuar modificações no texto, à posteriori, será utilizado outro pequeno monitor.

NOTA: Claro que também pode modificar um texto com este mesmo programa, voltando a escrevê-lo de novo.

```

9900 BORDER 5: PAPER 5: INK 1: C
LS : PRINT AT 8,2: INK 0;"INTROD
UCAO DE TEXTO NA REM 0": PAUSE 2
00: CLS
9905 PRINT " INK 1;" Instruções
"" Este programa parará sempre
e que a extensão dos dados entrad
os seja superior ao número de
bytes reservados pelo monitor ""1
"" Este monitor serve apen
as para entrada de texto e grupo
s de números, a serem utilizados
por rotinas em código máquina."
9915 PRINT " Em texto, tenha at
enção ao espaçamento e separação
das palavras de forma a obter gr
upos exactos de 32 caracteres."
9920 LET x=23760
9925 PRINT " INK 0;" Para fim d
e mensagem prima "#" (código 35
)"
9930 INPUT " Introduza uma linha
de dados. ";a$
9940 FOR n=1 TO LEN a$
9945 IF PEEK x=13 THEN STOP
9950 POKE x,CODE a$(n): IF n=LEN
a$ THEN GO TO 9975
9955 PRINT CHR$ CODE a$(n)
9960 LET x=x+1
9965 NEXT n
9970 GO TO 9930
9975 IF CODE a$(n)=35 THEN STOP
9985 PRINT CHR$ CODE a$(n)
9990 GO TO 9960
9999 SAVE "2" LINE 9900

```

PROGRAMA "3" — Introduzidor de C.M. na REM 0 em código decimal.

Este monitor tem as mesmas características técnicas dos anteriores e deve ser chamado com a instrução MERGE "3". Tem instruções de utilização e só deve ser usado para código máquina.

Cada vez que introduzir um código, aparece no écran, por uma instrução PEEK X, a confirmação deste cón-

go e o respectivo endereço. Se verificar que se enganou prima "E" e volte o número.

O monitor faz STOP, quando:

- Terminar a REM.
- Meter como código um número superior a 255.

```

9900 BORDER 5: PAPER 5: INK 0: C
LS: PRINT AT 2,1;"Introducao de
Codigo Maquina na REM 0."
9910 PRINT "" Entre com o Codig
o em DECIMAL."
9915 PRINT "" Para TERMINAR use
o numero 255."
9920 PRINT "" Se ERROU o Codigo
e so reparou depois de ENTER, p
rima a tecla E"
9925 PRINT "" Prima uma tecla p
ara COMECAR."
9930 INPUT l$: CLS
9940 LET x=23760
9945 INPUT "codigo entre 0 e 255
: ";a$: IF a$="E" OR a$="e" THEN
GO TO 9935
9950 LET i=VAL a$
9955 IF i>255 OR PEEK x=13 THEN
STOP
9960 POKE x,i
9965 PRINT ,TAB 2;x,i
9970 LET x=x+1
9975 GO TO 9945
9985 LET x=x-1: INPUT "Meta o co
digo CERTO ";i
9990 GO TO 9955
9999 SAVE "3" LINE 9900

```

PROGRAMA "4" — Correção de erros ou alterações do texto e transferência do conteúdo da REM 0, para qualquer ponto da RAM.

Este monitor deve ser chamado por MERGE "4". Ao correr, apresenta o MENU com 2 opções.

Para emendar um erro ou substituir texto, selecione a 1.ª rotina, entrando com o número 1.

Surgem instruções e para continuar prima uma tecla. Cada vez que premir ENTER, avança um endereço e mostra o seu conteúdo. Para o alterar, prima a tecla "E" (erro), introduza o novo código e depois ENTER. Para transportar o conteúdo da REM 0, selecione a 2.ª rotina, entrando com o número 2. Surgem instruções e para continuar prima uma tecla.

Ao correr, esta rotina avisa o utilizador que deve proteger os bytes a gravar, alterando o RAMTOP. A transferência dos bytes termina quando o ciclo encontra o fim do texto "#" (código 35) ou ainda o código 13 (fim da REM).

Quando aparecer o relatório, pode se quiser, anular o programa monitor introduzindo NEW e ENTER.

O programa indica ainda como gravar esses bytes. E pronto caro leitor, programas monitores apenas escritos em Basic, terminam aqui. Alguns deles vão ser utilizados na nossa iniciação à linguagem máquina. Outros já em programação mista (Basic e C.M.), após a parte teórica das instruções assembler, irão surgir.

```

9900 BORDER 5: PAPER 5: INK 0: C
LS: PRINT AT 2,11;"M E N U : "
"" 1 - Emenda de erros, alteraca
de texto etc., na REM 0."
"" 2 - Transferencia do conteud
da REM 0, para outro pont
da RAM, com inicio no end
reco (x)."
9905 PRINT "" Indique o numer
o desejado." : INPUT n: CLS : IF
n=2 THEN GO TO 9950
9910 PRINT AT 3,11;"Rotina 1""
Instrucoes: "" Cada ENTER avan
ca um endereco e mostra o seu co
teudo. Para alterar-to prima "E"
, introduza novocodigo e ENTER."
"" Quando acabar as correccoe
s, prima "F".""" "Prima uma tec
la." : INPUT l$: CLS : LET x=2376
0
9915 LET a=PEEK x: IF a=13 THEN
STOP
9920 IF a>31 THEN PRINT ";x,CHR$
a)
9925 IF a<32 THEN PRINT ";x,"Cod
igo ";a)
9930 INPUT l$: IF l$="F" OR l$="
F" THEN STOP
9940 IF l$="E" OR l$="e" THEN IN
PUT a$: LET k=CODE a$: POKE x,k:
PRINT "" ;CHR$ PEEK x
9945 LET x=x+1: GO TO 9915
9950 PRINT AT 0,11;"Rotina 2""
Instrucoes: "" Se ainda nao o
rez deve alterara RAMTOP para (n
-1). "" A transferencia de byt
es acabaquando aparecer no texto
o codigo 35 "#" ou o codigo 13
, que indica o fim da REM. "" Q
uando surgir o relatorio, podean
ular o Basic, com NEW e ENTER."
"" Para gravar os bytes transfe
ridos, escreva: SAVE ""nome"" C
ODE (endereco n), (numero de byte
s). "" "Prima uma tecla": INPUT
l$: CLS
9955 INPUT " Ja tem RAMTOP? (s o
u n)";r$: IF r$="n" THEN PRINT "
Depois de a alterar, prima RUN.
": STOP
9960 LET x=23760: INPUT " Indiqu
e o endereco de arranque para a
transferencia. ";n
9965 POKE n,PEEK x: PRINT " ";x)
"" ;PEEK x,n) "" ;PEEK n
9970 LET x=x+1
9975 LET n=n+1
9985 IF PEEK x=35 OR PEEK x=13 T
HEN STOP
9990 GO TO 9965
9999 SAVE "4" LINE 9900

```

Neste computador, toda a programação em BASIC, pode ser substituída por código máquina, sem qualquer limitação.

Temos no entanto de tomar consciência que, para o fazer, é necessário um conhecimento profundo do microprocessador, do seu assembler e das rotinas monitoras da ROM, pelo que esse estudo terá de ser cuidadoso e bastante ilustrado com programas exemplificativos para dele tirarmos proveito.

Antes de terminarmos o primeiro capítulo, 2 programas, um em Basic e outro em C.M. (nesta ocasião do curso, inscritos como mera demonstração) vão esclarecer o leitor sobre a razão fundamental dos mais

experientes insistirem na programação em código máquina.

Ambos os programas conseguem, com igual eficiência mas em tempos muito diferentes, misturar duas imagens numa única.

Isto é uma coisa muito importante, que os ingleses chamam "MERGE PICTURES", e de uma utilidade fora de série.

O princípio do seu funcionamento é muito simples. Temos uma imagem no Ecran, e com um comando RAND USR XXXX obtemos uma sobreposição dum desenho, dum texto, de legendas, etc., que se encontram depositados algures na RAM.

Uma simples operação binária "OR" é a *chave* do programa.

Em código máquina, como terão oportunidade de ensaiar, a execução é instantânea; mas em Basic demora ± 2 horas a completar o ecran.

```

5 BORDER 5: PAPER 5: INK 1: C
LE : LIST : STOP
6 REM MERGE o ecran guardado
no endereço 26000, com o ecran
presente
7 REM Programa de demonstracao
em Basic
8 REM desenha a figurae meta-
a na memoria com GOTO 400
9 REM Para Juntar os ecrans G
OTO 10
10 LET hl=16384
20 LET de=26000
30 LET bc=6144
40 LET a=PEEK de
50 GO SUB 1000
60 POKE hl,a
70 LET hl=hl+1
80 LET de=de+1
90 LET bc=bc-1: IF bc=0 THEN S
TOP
100 GO TO 40
400 LET x=26000
410 LET z=16384
420 LET y=6144
430 POKE x,PEEK z
440 LET x=x+1
450 LET z=z+1
460 LET y=y-1: IF y=0 THEN STOP

```

```

470 GO TO 430
1000 REM subrotina que executa a
operacao binaria (a OR PEEK hl)
1005 REM decimal em binario
1010 RESTORE 1140: LET k=0: LET
m=0
1020 LET b=PEEK hl
1030 FOR n=7 TO 0 STEP -1
1040 READ f: LET c=INT (b/f)
1050 LET a=a-c*f
1060 LET d=INT (b/f)
1070 LET b=b-d*f
1080 LET k=k+(c OR d)*10^n
1090 NEXT n
1095 REM binario em decimal
1100 RESTORE 1140: LET n#=STR$ k
1110 FOR n=LEN n#-1 TO 0 STEP -1
1120 READ f: LET m=m+f*VAL n#(LE
N n#-n)

```

```

1130 NEXT n
1135 LET a=m
1140 DATA 128,64,32,16,8,4,2,1:
REM (2^n)
1499 RETURN
9999 SAVE "MERGEECRAN" LINE 5

```

```

10 REM MERGE ECRAN programado
em código máquina
15 REM introducao do CM
20 CLEAR 25699: LET x=32700
30 FOR n=x TO x+21
40 READ a: POKE n,a
50 PRINT n,PEEK n
60 NEXT n
62 STOP
65 REM *****
100 REM Transporte d'ecran para
a memoria, programado em Basic
140 LET y=16384: LET z=26000: L
ET k=6144
150 POKE z,PEEK y
160 LET z=z+1: LET y=y+1: LET k
=k-1
170 IF k=0 THEN STOP
180 GO TO 150
200 REM *****
210 REM Merge em cm
220 LET e=23296
230 POKE e,144: POKE e+1,101
240 RANDOMIZE USR 32700
250 STOP
300 DATA 33,0,64,237,91,0,91,1,
0,24,26,182,119,35,19,11,120,177
,32,246,201
9999 SAVE "MERGEECRAN" LINE 10

```

32700	33
32701	0
32702	64
32703	237
32704	91
32705	0
32706	91
32707	1
32708	0
32709	24
32710	26
32711	182
32712	119
32713	35
32714	19
32715	11
32716	120
32717	177
32718	32
32719	246
32720	201

Vale a pena estudar código máquina, não vale? Eu, caro leitor, direi que SIM...

FIM DO 1º CAPÍTULO

(Cont. no próximo número)

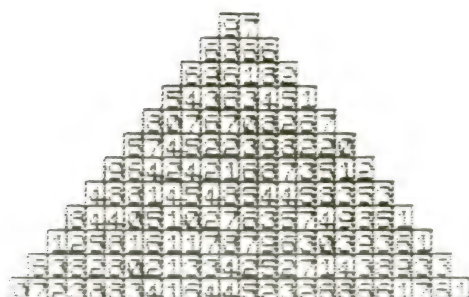
PIRAMIDE

SPECTRUM 16 K

In. PERSONAL COMPUTER, NOV./83

Adapt. e Trad. de J. MAGALHÃES

PIRAMIDE



MEU TOTAL 48

TEU TOTAL 7

PUZZLE

DESENHO A. PIRAMIDE CHEIA DE
NÚMEROS, SIGO UMA TRAJETÓRIA
AT ACASO, O MEU TOTAL É A SOMA
DOS NÚMEROS DAS CASAS POR ONDE
PASSEI (DO TOPO À BASE)

TEM DE ENCONTRAR O MEU CAMINHO,
OU UM OUTRO, MAS NESTE CASO COM
UM TOTAL IGUAL AO MEU, OU SEJA,
A SOMA DAS CASAS POR ONDE PASSAR
DEVERÁ SER IGUAL À MINHA.

```

10 GO SUB 2000
90 BORDER 0: INK 0: PAPER 2: C
LS
100 REM LINHAS HORIZONTAIS
105 LET B=223
110 LET C=191
115 PLOT 223,56: DRAW -191,0
120 FOR A=64 TO 152 STEP 8
130 PLOT B,A
140 DRAW -C,0
150 LET B=B-8
160 LET C=C-16
170 NEXT A
200 REM LINHAS VERTICAIS
205 LET B=56
210 LET A=127
215 LET C=96
220 PLOT 127,56: DRAW 0,96
230 FOR D=8 TO 96 STEP 8
240 PLOT A+D,B
250 DRAW 0,C
260 PLOT A-D,B
270 DRAW 0,C
280 LET C=C-8
290 NEXT D

```

```

300 REM
310 RANDOMIZE 0:
315 LET D=14
320 LET N=2
330 DIM X(12,24)
340 FOR A=1 TO 12
350 FOR B=1 TO N
360 LET X(A,B)=INT ((RAND*9)+.5)
370 PRINT OVER 1: PAPER 7: INK
1,AT A+2,D+B;X(A,B)
380 NEXT B
385 LET D=D-1
390 LET N=N+2
395 NEXT A
400 REM
410 LET B=1: IF RAND+.5 THEN LET
B=2
415 LET I=B
420 DIM H(12): DIM U(12)
430 LET TOTAL=X(1,B)
440 LET U(1)=3: LET H(1)=B+14
450 FOR A=2 TO 12: LET U(A)=A+2
460 LET Z=INT ((RAND*9)+.5)
470 IF Z=3 THEN LET Z=-1: GO T
O 500
480 IF Z=6 THEN LET Z=0: GO TO
500
490 LET Z=1
500 LET B=B+Z+1
510 LET TOTAL=TOTAL+X(A,B)
520 LET H(A)=15-A+B
530 NEXT A
550 REM
560 GO SUB 1500
600 REM
610 LET B=I
615 DIM D(12): DIM E(12)
620 GO SUB 3000
630 LET GUESS=X(1,B)
640 LET E(1)=3: LET D(1)=B+14
645 PRINT OVER 1: PAPER 5:AT E(
1),D(1):" "
650 FOR A=2 TO 12
655 BEEP .5,40
660 PRINT PAPER 0: INK 5:AT 19,
29:GUESS
670 LET E(A)=A+2
675 REM
680 IF INKEY$="" THEN GO TO 680
690 IF INKEY$=CHR$ 53 THEN LET
Z=-1: GO TO 750
700 IF INKEY$=CHR$ 54 THEN LET
Z=0: GO TO 750
710 IF INKEY$=CHR$ 56 THEN LET
Z=1: GO TO 750
720 IF INKEY$=CHR$ 82 OR INKEY$
=CHR$ 114 THEN GO SUB 1600: GO S
UB 1500: GO TO 600
730 IF INKEY$=CHR$ 83 OR INKEY$
=CHR$ 115 THEN GO TO 900
740 GO TO 680
750 LET B=B+Z+1
760 LET GUESS=GUESS+X(A,B)
770 LET D(A)=15-A+B
780 PRINT OVER 1: PAPER 3:AT E(
A),D(A):" "
790 NEXT A
900 REM
910 PAUSE 1
1000 REM
1010 FOR A=1 TO 12
1020 PRINT OVER 1: PAPER 4:AT U(
A),H(A):" "
1030 NEXT A
1035 REM
1040 FOR B=13 TO 20
1050 FOR A=1 TO 30

```



```

1060 PRINT PAPER 2; AT B,A; " "
1070 NEXT A: NEXT B
1075 REM
1080 PRINT INK 0; AT 0,5; "ESTE FO
I O MEU CAMINHO"
1090 LET A=17: FOR B=1 TO 30: PR
INT PAPER 2; INK 1; AT A,B; "█"; AT
A+4,B; "█" NEXT B
1100 FOR A=18 TO 20: FOR B=1 TO
5: PRINT INK 1; AT A,B; "█"; AT A,2
5+B; "█" NEXT B: NEXT A
1110 IF GUESS=TOTAL THEN GO TO 1
300
1200 REM
1210 PRINT FLASH 1; BRIGHT 1; OV
ER 1; INK 1; AT 19,9; " TOTAL ERRA
DO"
1220 GO TO 1400
1300 REM
1310 BEEP .15,5: BEEP .15,10: BE
EP .15,15: BEEP .15,22: BEEP .15
,17: BEEP .15,22
1320 PRINT FLASH 1; BRIGHT 1; OV
ER 1; INK 1; AT 19,10; " CORRECTO
"

```

```

1400 REM
1410 PAUSE 250: PRINT OVER 1; IN
K 7; AT 19,1; "TECLA"; AT 19,26; "EN
TER"
1420 PAUSE 0
1430 PAPER 0: CLS
1450 PRINT INK 4; AT 11,7; "TECLA
""S"" PARAR"; AT 13,3; "QUALQUER
TECLA REPETIR": PAUSE 0
1470 IF INKEY$=CHR$ 83 OR INKEY$
=CHR$ 115 THEN CLEAR: BORDER 7:
PAPER 7: INK 0: CLS: STOP
1480 CLEAR: GO TO 1
1490 STOP
1500 REM
1510 FOR A=1 TO 12 STEP DIFF+1
1520 PRINT OVER 1; PAPER 4; AT U(
A),H(A); " "
1530 NEXT A
1540 RETURN
1600 REM
1610 FOR A=2 TO A-1
1620 PRINT OVER 1; PAPER 7; AT E(
A),D(A); " "
1630 BEEP .25,50
1640 NEXT A

```

COLONY INVADERS

SPECTRUM 16 K

In. PERSONAL COMPUTER, DEZ./83

Adapt. e Trad. de J. MAGALHÃES

COLONY INVADERS

Tem o controlo de um dispositivo criado para evitar a intrusão de naves espaciais que atiraves de raio que lançam sobre a terra, destroem todo o tipo de vegetação.

Tem apenas 3 tentativas para não permitir a aproximação dessas naves do sol, que também vai sendo afectado. Cada planta que seja destruída corresponde a menos uma vida ou seja a uma tentativa.

TECLAS DE CONTROLO

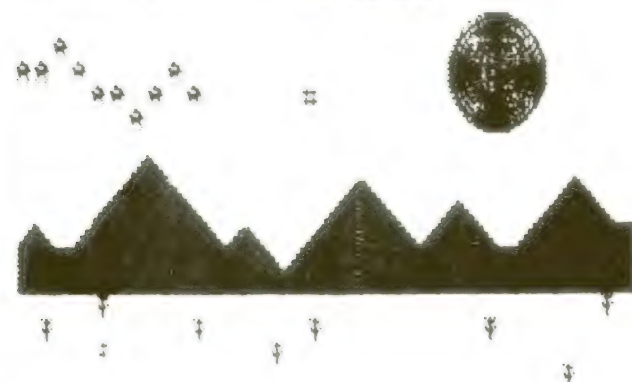
"5" - Esquerda "6" - Descer

"7" - Subir "8" - Direita

Procure controlar o "#" sobre a nave e o disparo será automático

Score: 0

High: 0



```

1 LET high=0
10 DATA BIN 000000001 BIN 00000
011 BIN 000001111 BIN 000011111 BI
N 000111111 BIN 001111111 BIN 0111
1111 BIN 111111111
20 DATA BIN 100000000 BIN 11000
000 BIN 111000000 BIN 111100000 BI
N 111110000 BIN 111111000 BIN 1111
1110 BIN 111111111
30 FOR X=0 TO 7: READ Y: POKE
USR "5"+X,Y NEXT X
40 FOR X=0 TO 7: READ Y: POKE
USR "6"+X,Y NEXT X
50 BORDER 0 PAPER 5 INK 1: CLS
60 PRINT AT 12,0; "

```



70 PRINT PAPER 4; "

```

80 FOR X=0 TO 20: CIRCLE INK 5
,200,140,X NEXT X
90 DATA BIN 00011000 BIN 00111
100 BIN 01100110 BIN 01111110 BI
N 00111100 BIN 00100100 BIN 0100
0010 BIN 00100100
95 DATA BIN 00101010 BIN 00011
100 BIN 00001000 BIN 00101010 BI
N 00011100 BIN 00001000 BIN 0000
1000 BIN 00001000
97 FOR X=0 TO 7: READ Y: POKE
USR "C"+X,Y NEXT X
100 FOR X=0 TO 7: READ Y: POKE
USR "d"+X,Y NEXT X
101 PRINT INK 6 PAPER 4; AT 19,
1; "X"; AT 20,4; "X"; AT 19,15; "X"; AT
19,9; "X"; AT 15,4; "X"; AT 20,13;
"X"; AT 19,24; "X"; AT 18,30; "X"; AT
21,28; "X"
103 LET c=15: LET d=5: LET scor
e=0: LET vida=9
104 PRINT AT 0,0; "Score: " ; scor
e; AT 0,15; INVERSE 1; "High: " ; hi
gh
105 LET X=5: LET Y=0
107 LET X=(INT (RAND*31)-1)+X: IF

```



```

X>7 THEN LET X=X-2: GO TO 107
108 IF X<3 THEN LET X=X+2: GO T
O 107
110 IF ATTR (X+1,Y+2)=46 OR ATT
R (X-1,Y+2)=46 OR ATTR (X,Y+1)=4
6 THEN GO TO 200
115 GO SUB 500
117 IF X=d AND Y=c THEN LET sco
re=score+(25-Y): PRINT AT 0,6,sc
ore: BEEP .2,10: BEEP .3,20: GO
TO 105
120 PRINT INK 2:AT X,Y:"@"
130 BEEP .1,X*5: PRINT AT X,Y:"
": LET Y=Y+1: GO TO 107
200 PRINT AT X,Y: BRIGHT 1: INK
2: PAPER 5:"*": READ e,f: PRINT
AT e,f: INK 3: PAPER 4:"*":
210 FOR t=0 TO 1 STEP .05: BEEP
.01,t: NEXT t: FOR t=1 TO 0 STE
P -.05: BEEP .01,t: NEXT t
220 PRINT AT X,Y: PAPER 5,"*":
$
240 LET vida=vida-1: IF vida=0
THEN GO TO 1000
245 DATA 19,1,20,4,19,15,19,9,1
8,4,20,13,19,24,18,30,21,28
250 GO TO 105
500 PRINT AT d,c:" "
505 IF INKEY$="5" THEN LET c=c-
1: IF c<0 THEN LET c=0
510 IF INKEY$="6" THEN LET d=d+
1: IF d>7 THEN LET d=7
520 IF INKEY$="7" THEN LET d=d-
1: IF d<3 THEN LET d=3
530 IF INKEY$="8" THEN LET c=c+
1: IF c>20 THEN LET c=20
550 IF INKEY$=CHR$ 8 THEN LET c
=c-2: IF c<0 THEN LET c=0
560 IF INKEY$=CHR$ 9 THEN LET c
=c+2: IF c>20 THEN LET c=20
570 IF INKEY$=CHR$ 10 THEN LET
d=d+2: IF d>7 THEN LET d=7
580 IF INKEY$=CHR$ 11 THEN LET
d=d-2: IF d<3 THEN LET d=3
590 PRINT AT d,c: INK 7,"#"

```

```

600 RETURN
1000 RESTORE
1010 IF high(score) THEN LET high
=score
1020 PRINT AT 10,10: FLASH 1: IN
K 0: OVER 1: PAPER 3:"PLANTAS OU
EIMADAS":AT 11,10," POR RADIACA
O
1030 FOR e=1 TO 1000: NEXT e
1040 GO TO 9

```

NOTAS SOBRE O PROGRAMA

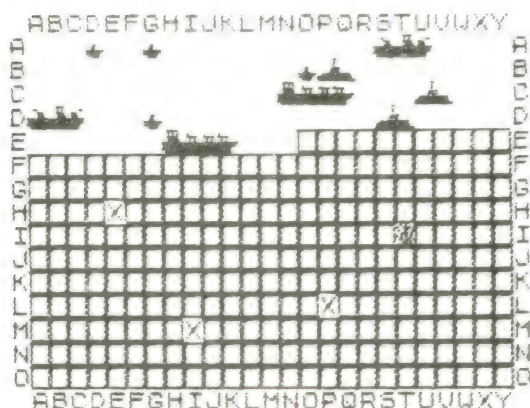
10-40	Definicao de graficos
50-70	Apresentacao do jogo no ecran
80-101	Formacao do sol Definicao do caracteres Graficos "c" e "d"
103-105	Estabelece as variaveis do jogo; score e High score, as coordenadas X,Y das naves.
107-115	Atribuicao de um valor ao acaso para X; verifica a aproximacao das naves do sol.
117-220	Destruido do invasor alteracao do score; novo invasor; morte de uma planta.
240-500	Numero de vidas; DATA que determina a ordem de destruicao das plantas; teclas de controle; dispositivo de disparo.
1000-1040	Alteracao do score max.; fim do jogo.

BATALHA NAVAL

SPECTRUM 16K




Adapt. de ANTÓNIO AMARAL
Porto

B A T A L H A N A V A L



Coord. hor.?

PONTUAÇÃO

* = 20 Pontos.
 = 40 Pontos.
 = 60 Pontos.
 = 100 Pontos.

Tem de localizar 20 inimigos.

```

1 REM "B. Naval"
10 BORDER 1: PAPER 1: INK 7: C
LS
20 LET z$=" B A T A L H A "
30 LET z$=z$+" N A V A L "
40 LET END=2000
90 FOR a=1 TO LEN z$
92 IF z$(a)=" " THEN PRINT " "
: NEXT a
94 BEEP .01,CODE z$(a)-100: PR
INT z$(a): NEXT a
100 FOR a=144 TO 154: FOR b=0 T

```

QUALQUER TECLA


```

0 7: BEEP .01,30: READ C: POKE U
SR CHR$ a+b, C: NEXT b: NEXT a: P
OKE USR CHR$ 156,255: FOR a=1 TO
6: POKE USR CHR$ 156+a,129: NEX
T a: POKE USR CHR$ 156+7,255
110 POKE 23692,255: PRINT AT 21
,0:
120 PRINT "*****INSTRUÇÕES*****"
mece por dar entrada das duas co
ordenadas." (A-Y) "Seguindo-se
as coordenadas." (A-O) " (
SPACE) para desistir."
130 PRINT "*****QUALQUER T
ECLA PAUSE 0
140 PRINT "*****PONTUAÇÃO*****"
= 20 Pontos. " = 40 Pon
tos. " = 60 Pontos. " = 80
= 100 Pontos. " Tem de localiz
ar 20 inimigos.
160 DIM a$(15,25)
162 FOR A=1 TO 3
164 GO SUB 200: IF a$(X,Y TO Y+
3)<>" THEN GO TO 164
170 LET a$(X,Y TO Y+3)="":
NEXT A
172 FOR a=1 TO 5
174 GO SUB 200: IF a$(X,Y TO Y+
2)<>" THEN GO TO 174
176 LET a$(X,Y TO Y+2)="": N
EXT a
178 FOR a=1 TO 9
180 GO SUB 200: IF a$(X,Y TO Y+
1)<>" THEN GO TO 180
182 LET a$(X,Y TO Y+1)="": NE
XT a
184 FOR a=1 TO 5
186 GO SUB 200: IF a$(X,Y)<>" "
THEN GO TO 186
188 LET a$(X,Y)="": NEXT a
190 GO TO 300
200 LET X=INT (RND*15)+1: LET Y
=INT (RND*22)+1: RETURN
300 PRINT AT 21,0: "Qualquer tec
la para começar": PAUSE 0: PAPER
5: FOR a=1 TO 22: PRINT "
": NEX
T a: BORDER 5: PAPER 5: INK 0: C
LS
310 FOR a=1 TO 15: PRINT AT a,1
, " ": NE
XT a
320 POKE 23658,8: LET HIT=0: LE
T amo=300: LET s=0: LET x=1: LET
y=1
330 FOR a=65 TO 89: PRINT AT 0,
a-64: PAPER 7: CHR$ a: AT 16,a-64:
CHR$ a: NEXT a
340 FOR a=65 TO 79: PRINT AT a-
64,0: PAPER 7: CHR$ a: AT a-64,26:
CHR$ a: NEXT a
350 PRINT #1, AT 1,1: " SCORE=";s
, " AT 1,17: " TIROS=";AMO; "
352 IF HIT=46 THEN GO TO 2500
355 PRINT AT 20,1: "Coord. hor.?"
: LET b$=INKEY$: IF b$="" THEN
GO TO 350
357 IF INKEY$="" THEN GO TO EN
D
360 IF CODE b$>89 OR CODE B$<65
THEN GO TO 350
370 LET y=(CODE b$)-64: PRINT A
T 20,15;b$
380 BEEP .1,6
385 IF INKEY$<>" " THEN GO TO 38
5
390 PRINT AT 21,1: "Coord. ver.?"
: LET b$=INKEY$: IF b$="" THEN
GO TO 390
395 IF INKEY$="" THEN GO TO EN
D
400 IF CODE b$>79 OR CODE B$<65

```

```

THEN GO TO 390
410 LET x=(CODE b$)-64: PRINT A
T 21,15;b$
420 PRINT AT X,Y: "X"
430 GO SUB 1000
440 PRINT AT 20,15: " AT 21,1;
": GO TO 350
1000 IF amo<=0 THEN PRINT #1, AT
1,7: FLASH 1: "n. de tiros esgota
do": BEEP 1,-6: FOR A=-6 TO -30
STEP -1: BEEP .05,A: NEXT A: GO
TO 2000
1010 FOR a=5 TO -30 STEP -5: BEE
P .03,a: PAUSE 10: NEXT a
1020 LET amo=amo-1: IF a$(x,y)="
" THEN RETURN
1030 LET HIT=HIT+1: PRINT AT x,y
: PAPER 7: BRIGHT 1;a$(x,y)
1040 LET point=20:
1050 IF a$(x,y)=" " OR a$(x,y)="
" OR a$(x,y)=" " OR a$(x,y)=" "
THEN LET point=25
1060 LET s=s+point
1070 FOR B=-40 TO -30: FOR A=0 T
O 7: BORDER A: BEEP .03,B: NEXT
A: NEXT B: PRINT AT X,Y: FLASH 1
: PAPER 2: INK 6: "
1080 IF HIT=46 THEN GO TO 2500
1100 BORDER 5: RETURN
2000 PRINT #1, AT 0,0: FLASH 1: "
Quer mesmo desistir?
S or N
2010 IF INKEY$="S" OR INKEY$="s"
THEN GO TO 2100
2015 IF INKEY$="N" OR INKEY$="n"
THEN BEEP 1,30: PRINT #1, AT 0,0
: PAPER 5: "
": GO TO 350
2020 GO TO 2010
2100 FOR A=1 TO 15: FOR B=1 TO 2
5: BEEP .05,A+B: PRINT AT A,B: P
APER 5: INK 0;A$(A,B): NEXT B: N
EXT A
2110 GO TO 3000
2500 FOR A=1 TO 5: FOR B=1 TO 7:
BORDER B: BEEP .01,A*10: NEXT B
: NEXT A
2510 FOR A=16 TO 21: PRINT AT A,
0: "
": NEXT A: PRINT #1, AT 0,0: "
"
2520 PRINT AT 17,2: "PARABENS CON
SEGUIU ";S; " PONTOS"
COM UM
BONUS DE ";AMO;" PONTOS"
3
OMANDO ASSIM ";AMO+5;" PONTOS"
2550 GO TO 3000
3000 PRINT #1, AT 0,0: "NOVO JOGO?
S=SIM N=NAO
3010 IF INKEY$="S" OR INKEY$="s"
THEN POKE 23692,255: GO TO 120
3020 IF INKEY$="N" OR INKEY$="n"
THEN POKE 23692,255: GO TO 3050
3030 GO TO 3010
3050 FOR A=0 TO 5: FOR B=0 TO 7:
BEEP .01,A*5: BORDER B: PAPER B
: CLS: NEXT B: NEXT A
3060 INK 0: PRINT " OK. Obri
gado pelo seu jogo
B A T A
L H A N A U A L " "Este program
a sera auto-
destruido em
10 s."
3070 PAUSE 10: PRINT AT 10,9: "
: FOR a=9 TO 0 STEP -1: PRINT AT
10,19;a: BEEP .01,-20: PAUSE 40
: NEXT a

```



```

3100 NEW
9000 DATA 0,0,0,24,24,255,126,60
,40,157,73,58,170,109,26,255
9010 DATA 1,1,1,3,3,63,127,255,1
28,0,0,192,192,255,254,255
9020 DATA 0,48,14,14,255,255,255
,127,15,10,15,15,255,255,255,255
,0,6,56,56,255,254,252,248
9030 DATA 62,42,62,62,255,255,25
5,127,0,32,62,20,255,255,255,255
,0,65,126,40,255,255,255,255,0,0
,240,160,255,252,248,240
9999 SAVE "B.Naval" LINE 1: BORD
EA 0 PAPER 0: CLS : BEEP .1,6:
VERIFY ""

```

DEFINIR GRAFICOS

```

+ GRAPHICS + TECLA "A"
+ " + " "C+D"
+ " + " "E+F+G"
+ " + " "H+I+J+K"
+ " + " "B"
+ " + " "M"

```

NOTE:

So poderá ler os graficos definidos depois de passado todo o programa.

FREQUÊNCIAS

SPECTRUM

Autor: Silvestre Carneiro

Este programa mede a frequência de sinais entrados pela ficha EAR. Experimente, por exemplo, uma cassete com música.

```

5 REM "Freq."
10 REM Mede frequencias ate 30k
Hz; input no jack EAR
20 GO TO 100
30 PRINT AT 10,5;"FREQ.= ";PEE
K 32768+256*PEEK 32769;" Hz
40 RANDOMIZE USA 32000
50 PAUSE 50: GO TO 30
100 CLEAR 31999
110 FOR f=32000 TO 32038
120 READ a: POKE f,a: NEXT f
130 GO TO 30
140 DATA 58,120,92,71,58,120,92
,184,40,250,198,50,71,33,0,0,219
,254,203,119,40,250,35,219,254,2
03,119,32,250,34,0,128,58,120,92
,184,200,24,233

```

CASA ASSOMBRADA

SPECTRUM 16K

Adapt. de ANTÔNIO AMARAL
Porto

```

4 CLEAR 34999: LET ns=0: LET
h$="X" PRINT AT 10,5: FLASH 1
:"ESPERE UM MOMENTO": GO TO 1000
5 IF ATTA (a+1,b)=6 THEN PRIN
T AT a+1,b: INK 4:"U"
6 RETURN
50 PRINT OVER 1: INK 2: PAPER
3:AT a,b:"X"
51 IF a=4 AND b=15 AND h=1 THE
N BEEP .1,10: LET t=t+5: IF t=2
0 THEN LET t=20: PRINT AT 2,0,"
": LET h=0: LET s=s+100: PRINT A
T 3,15: INK 6:"X"
52 LET a=a+(INKEY$="a" AND ATT
a (a+1,b)=2)-(INKEY$="r" AND ATT
a (a-1,b)=2)
53 IF a=12 AND b=26 AND ATTA (
a,b+1)=5 THEN BEEP .1,20: PRINT
AT 12,27: INK 7:"X": PRINT AT 2,
0:"X": LET h=1
54 LET b=b+(INKEY$="p" AND b<=
26 AND ATTA (a,b+1)<6)-(INKEY$=
"o" AND b=2 AND ATTA (a,b-1)<6
)
55 PRINT OVER 1:AT a,b: INK 9:
"X"
56 IF INKEY$="z" THEN GO SUB 5
57 PRINT AT 0,10-LEN STR$ s: I
NK 6:s
58 LET t=t-.015: IF t<=0 THEN
GO SUB 300
59 PRINT AT 21,6: INK 6: PAPER
2:AT t TO 1: PAPER 0:"X"
60 PRINT AT c,d: OVER 1: INK 2
:"X"
61 IF ATTA (c+1,d)=4 THEN POKE
35006,5: POKE 35025,28: RANDOMI

```

```

ZE USA 35000: PRINT AT c+1,d: IN
K 5: BRIGHT 1:"X": POKE 35025,29
RANDOMIZE USA 35000: PRINT AT
c+1,d: INK 5:"X": LET c=4: RANDO
MIZE: LET d=(RAND+10)+1: LET s=s
+25
64 LET d=d+(d<=b AND ATTA (c,d
+1)<6)-(d=b)
65 LET c=c-(c>=a AND ATTA (c-1
,d)<6 AND ATTA (c-1,d)<4)+(c<=
a AND ATTA (c+1,d)<6 AND ATTA (
c+1,d)<4)
66 PRINT OVER 1:AT c,d: INK 5:
"X"
67 IF a=4 AND b=26 AND g=0 THE
N LET j=j-1: LET g=1: PRINT AT 4
,23: INK 7,s:( TO j):"X": BEEP .
1,50: LET u=1
68 IF a=19 AND b=3 AND u=1 THE
N LET s=s+275: POKE 35056,5: RAN
DOMIZE USA 35000: LET u=0: LET g
=0: IF j=0 THEN GO TO 200
69 IF a=INT c AND b=INT d THEN
GO TO 300
100 GO TO 50
200 LET s=s+INT (t*50)
210 PRINT AT a,b:"X":AT c,d:"X"
215 PRINT AT 20,1: BRIGHT 1: FL
ASH 1:"X" B 0 N U 5
220 POKE 35006,100: RANDOMIZE U
SA 35000
240 LET a=19: LET b=3: LET c=4
LET d=26: LET h=h+1: LET j=3
250 LET t=20-h: IF t<=10 THEN L
ET t=10
260 PRINT OVER 1:AT c,d:"X"

```



```

270 PRINT AT 4,28;J$( TO J)
280 PRINT AT 20,1; FLASH 0;"
290 GO TO S1
300 PRINT AT a,b; OVER 1;"X";AT
c,d; OVER 1;"M"
301 PRINT AT 21,27;" "
310 FOR n=10 TO 2 STEP -2: BEEP
.1,n: PAUSE 1: NEXT n
315 PAUSE 7: BEEP .075,3: PAUSE
1: PRINT AT a,b;"X": BEEP .075,
3
320 PRINT AT a,b;" "
331 LET m=m-1: IF m=0 THEN GO
TO 900
340 PRINT AT 21,28; INK 4;m$( T
O m);" "
350 LET a=19: LET b=3: LET c=4:
LET d=26: LET g=0: PRINT AT 4,2
8;J$( TO J)
360 PRINT AT c,d; OVER 1;"M"
390 GO TO S1
399 REM FIM DO JOGO
900 IF s>=hs THEN LET hs=s: PRI
NT AT 5,0; INK 7: PAPER 2;"NOVO
SCORE MAX!!! TUA3 INICIAIS"
901 IF s=hs THEN FOR n=-50 TO 5
0 STEP 5: BEEP .1,n: NEXT n
902 IF s=hs THEN INPUT h$: IF L
EN h$>3 THEN GO TO 902
903 PRINT AT 21,28;" "
931 PRINT AT 21,27;" ": PRINT A
T 11,0; INK 6; PAPER 1; FLASH 1;
"
"
932 PRINT AT 15,0; INK 0; PAPER
5;"
NOVO JOGO
S / N
933 IF INKEY$="3" OR INKEY$="4"
THEN RESTORE 9927: GO SUB 9903:
GO SUB 9920: GO TO S1
934 IF INKEY$="n" OR INKEY$="N"
THEN PRINT USA 0
940 GO TO 920
1000 GO SUB 9990: REM SOM
1010 GO SUB 9900: REM VARIAVEIS
1020 GO SUB 9800: REM GRAFICOS
1030 GO SUB 9700: REM INFO
1040 GO SUB 9920: REM SCREEN
1050 GO TO S1: REM JOGO
9699 REM INSTRUÇÕES
9700 BORDER 1: PAPER 1: INK 7: C
LS
9710 LET i$=" Bem vindo a casa
assombrada...Tera de passar po
r todas asdivisões da casa
e apanhar asjarras que se
encontram naultima divisao. Cad
a jarra, deveras colocada na car
rinha.
A impedir o seu obj
ectivo tem:
M - Fantasma
X - Skeleton
...O fantasma pode
ser evitadofazendo um laco.
O skeleton,usando o martelo.
TECLAS DE CONTROLO:
O - Subir
- Descer
- Direita
Z - Fazer o laco"
A
P
A
9717 PRINT "C A S A A A S O M
B A R A D A"
9718 PRINT " = = = = =
= = = = =
9720 FOR n=1 TO LEN i$: BEEP .00
5,s: PRINT i$(n): NEXT n
9725 POKE 35004,255: POKE 35006,
50: RANDOMIZE USA 35000
9730 PRINT #0; FLASH 1;" QUALQU

```

```

ER TECLA PARA COMECAR "
9740 PAUSE 0
9750 POKE 35025,29: RANDOMIZE US
A 35000
9760 RETURN
9799 REM U.D.G. 15
9803 FOR y=65368 TO 65479
9810 READ z: POKE y,z: NEXT y
9819 REM DATA U.D.G. 15
9820 DATA 56,40,145,124,56,56,40
,108,129,255,129,255,129,255,129
,255
9830 DATA 62,127,73,73,127,127,8
5,85,127,34,65,65,65,65,65,62
9840 DATA 8,93,73,62,8,20,34,34,
247,247,0,0,0,0,0
9850 DATA 0,24,56,112,104,4,2,0,
51,51,51,51,51,51,51,51
9860 DATA 0,0,0,0,15,7,7,31,0,0,
0,0,255,254,254,254,39,71,127,12
7,127,255,40,16,254,254,254,254,
254,255,20,8
9870 DATA 129,195,129,195,129,19
5,129,126,0,73,42,0,99,0,42,73
9890 RETURN
9899 REM VARIAVEIS
9900 LET t=20: LET t$="|||||||||||
||||||||||||||||"
9902 LET j=3: LET j$="000"
9903 LET a=19: LET b=3: LET c=4
LET d=26
9904 LET m$="XXX": LET m=3
9905 LET u=0: LET g=0
9907 LET h=0
9910 LET s=0
9918 RETURN
9919 REM SCREEN
9920 BORDER 0: PAPER 0: INK 7: C
LS: FOR n=5 TO 17 STEP 2: PRINT
AT n,0; INK 6;" ": NEXT n
9925 FOR n=1 TO 14: READ y,x: PR
INT AT y,x: INK 2;"|";AT y+1,x:
INK 2;"|";AT y-1,x: INK 2;"|": N
EXT n
9927 DATA 5,6,5,22,7,2,7,17,7,29
,9,15,11,5,11,24,13,3,13,17,15,1
3,15,30,17,7,18,7
9930 FOR n=1 TO 11: READ y,x: DR
AW INK 6;y,x: NEXT n
9932 DATA 0,144,127,31,1,0,127,-
31,0,-144,-254,0,0,144,127,31,12
6,-32,0,-142,-254,0
9933 PLOT 0,0
9934 PLOT 250,146: FOR n=1 TO 3:
READ x,y: DRAW INK
6;x,y: NEXT n
9935 DATA 0,8,-18,0,0,-4
9937 PLOT 246,154: FOR
n=1 TO 7:
READ x,y: DRAW INK 6;x,y: NEXT
n
9939 DATA 0,3,2,0,-5,10,-4,0,-5,
-10,2,0,0,-3
9940 PRINT AT 18,1; INK 2;"-";A
T 19,1; INK 2;"-";
9942 PRINT AT 12,27; INK 5;"X"
9945 PRINT AT 4,28;J$( TO J)
9947 PRINT AT 21,1; INK 7;"TEMPO
": INK 6; PAPER 2;I$( TO t); PA
PER 0;" " INK 4;m$( TO m)
9949 PRINT AT 21,30;" "
9952 PRINT AT 0,0; INK 4;"1 - ";
INK 6;"00000"AT 0,21; INK 5;h$
INK 7;" 00000"
9953 PRINT AT 3,0; OVER 1; INK 6
;" "
9954 PRINT AT 4,26; OVER 1; INK
2;"M"

```



```

9955 PRINT AT 0,30-LEN STR$ h$;
INK 7:h$
9960 PRINT AT 3,15; INK 6; PAPER
0;"X"
9970 PRINT , INK 6; AT 18,8;" "A
T 18,8; INK 6;" "
9980 RETURN
9988 REM SOM
9989 REM SOM DA SIRENE
9990 DEF FN a(n)=(CODE a$(n)-48
AND CODE a$(n)<58)+(CODE a$(n)-5
5 AND CODE a$(n)>64)
9991 LET a$="F3111000260A3A485C1
F1F1F0EFFFFE10ED794310FE2520F41C1
520E8F8C9"
9992 FOR J=1 TO LEN a$ STEP 2
9993 POKE 35000+J/2,16*FN a(J)+F
N a(J+1): NEXT J
9994 RETURN
9995 REM A B C D E F G H I J K
9996 REM A B C D E F G H I J K
9997 REM
9998 REM L M N
9999 REM L M N

```

C A S A A S S O M B R A D A
 = = = = =

Bem vindo a casa assombrada...
 Tera de passar por todas 25
 divisoes da casa e apanhar as
 jarras que se encontram na
 ultima divisao. Cada jarra, deve
 ser colocada na carrinha.

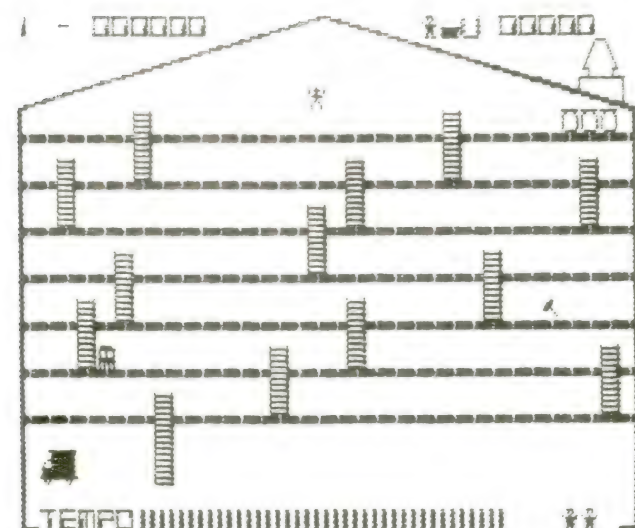
A impedir o seu objectivo tem:

☠ - Fantasma
 * - Skeleton

...O fantasma pode ser evitado
 fazendo um lado. O Skeleton
 usando o martelo.

TECLAS DE CONTROLO:

Q - Subir A - Descer
 D - Esquerda P - Direita
 Z - Fazer o lado



DEFINIR GRAFICOS

* - Graphics + tecla "A"
 * - " + " "E"
 * - " + " "C"
 * - " + " "I+J"
 * - " + " "F"
 * - " + " "G"
 * - " + " "H"
 * - " + " "n+L"
 * - " + " "B"
 * - " + " "n"
 * - " + " "m"
 * - " + " "D"

NOTE:

GRAPHICS- CAPS SHIFT + Tecla "S"

PERIGO NO ESPAÇO Zx81 16 K

FERNANDO PRECES

Sacavém

```

REM PROGRAMA MODIFICADO POR
ALMEIDA PRECES EM 25/6/1982
5 REM "4"
10 GOSUB 9000
15 RAND
18 CLS
20 LET MC=12
22 LET AI=0
25 LET AC=0
30 LET I=INT (21*RND)
40 LET J=INT (30*RND)
50 LET M=20
55 LET MI=0
60 LET N=20
70 PRINT AT I,J;" "
85 FOR L=1 TO (RAND*500)
86 NEXT L
90 GOSUB 2000
200 LET X$=INKEY$
210 LET Z$=X$
355 IF NOT X$="" AND RAND>(0.2+R
C/10) THEN GOTO 380
360 GOSUB 5000
365 IF MI=0 THEN GOTO 375
370 IF ABS (A-M)<2 AND ABS (B-
1)-N)<3 THEN GOSUB 7000
375 IF AI=0 THEN GOTO 385
380 IF ABS (A-I)<3 AND ABS (B-J
)<3 THEN GOTO 8000
385 IF Z$="Y" THEN GOTO 20
390 IF ABS (N-(J+1))<2 AND ABS
(M-I)<2 THEN GOTO 8500
400 IF X$="" THEN GOTO 700
420 IF X$="J" OR X$="K" OR X$="
N" OR X$="M" THEN GOSUB 4000
430 IF X$="A" OR X$="S" OR X$="
X" OR X$="Z" THEN GOSUB 3000
440 IF X$="L" THEN GOSUB 6000
580 FOR L=1 TO 5

```



```

535 NEXT L
600 GOTO 200
700 FOR L=1 TO 100
705 NEXT L
800 GOTO 200
2000 IF AI>0 THEN RETURN
2004 LET AI=1
2005 LET AC=AC+1
2010 LET A=1+INT (RAND*21)
2020 LET B=1+INT (RAND*29)
2030 PRINT AT A,B;"<■>"
2200 RETURN
3000 PRINT AT I,J;" "
3020 IF X$="2" THEN LET J=J-1
3030 IF X$="X" THEN LET J=J+1
3040 IF X$="5" THEN LET I=I-1
3050 IF X$="7" THEN LET I=I+1
3060 IF I=0 THEN LET I=30
3070 IF J=1 THEN LET J=1
3080 IF J=30 THEN LET J=30
3090 IF I=21 THEN LET I=21
3100 PRINT AT I,J;"■"
3900 RETURN
4000 IF MI=0 THEN RETURN
4020 PRINT AT M,N;" "
4030 IF X$="N" THEN LET N=N-2
4040 IF X$="K" THEN LET N=N+2
4050 IF X$="M" THEN LET M=M+2
4070 IF X$="J" THEN LET M=M-2
4080 IF M=21 OR M<0 OR N>31 OR N<0 THEN LET MI=0
4100 IF NOT MI=0 THEN PRINT AT M,N;"■"
4200 RETURN
5000 PRINT AT A,B;" "
5007 LET D=3+0.4*AC
5010 IF A<I THEN LET A=A-INT (D+
RAND)
5020 IF A>I THEN LET A=A+INT (D+
RAND)
5030 IF B<J THEN LET B=B-INT (D+
RAND)
5040 IF B>J THEN LET B=B+INT (D+
RAND)
5050 IF A=21 THEN LET A=21
5060 IF A=0 THEN LET A=0
5070 IF B=1 THEN LET B=1
5080 IF B=30 THEN LET B=30
5200 PRINT AT A,B;"<■>"
5300 RETURN
6000 IF MI>0 THEN RETURN
6011 IF MC=0 THEN RETURN
6012 LET MC=MC-1
6015 LET MI=1
6020 LET M=I+2
6030 IF M=21 THEN LET M=M-4
6040 LET N=J+2
6050 IF N=31 THEN LET N=N-4
6060 PRINT AT M,N;"■"
6070 RETURN
7000 PRINT AT 19,0;"NAVES INIMIG
AS DESTRUIDAS:";AC
7012 PRINT AT 20,0;"MISSEIS LANÇ
ADAS:";MC
7015 LET MI=0
7017 LET AI=0
7020 PRINT AT A,B;"■"
7025 PRINT AT M,N;"■"
7028 FOR L=1 TO 50
7029 NEXT L
7030 PRINT AT M,N;" "
7032 PRINT AT A,B;" "
7040 LET M$="EXPLOSAO PROXIMA.P
SUA NAVE FOI DESTRUIDA"
7050 IF ABS (J+1-N)<2 AND ABS (I
-M)<2 THEN GOTO 8800
7090 FOR L=1 TO 150
7091 NEXT L
7092 CLS
7094 PRINT AT I,J;"■"
7120 FOR L=1 TO (RAND*100)

```

```

7122 NEXT L
7150 GOSUB 2000
7200 RETURN
8005 LET M$="ELE APANHOU-O"
8010 GOTO 8800
8510 LET M$="O SEU MISSEL ATINGI
U-O"
8800 PRINT AT I,J;"■"
8850 FOR L=1 TO 100
8855 NEXT L
8860 PRINT AT 17,0;M$;" " TAB 0;
"O JOGO ACABOU.";"NAVES DESTRA.="
;"AC-1;" MISSEIS="IMO
8870 PRINT " QUER TENTAR DE NOVO
?"
8875 INPUT Z$
8876 IF CODE Z$=56 THEN GOTO 15
8888 STOP
9000 PRINT " BATALHA NO ESPACO
9005 PRINT "A SUA NAVE ESTA AV
ARIADA, VOCE"
9010 PRINT "NAO TEM CAMPO MAGNET
ICO EXTERIOR"
9015 PRINT "SE ATACADO DEVE FUGI
R, LANÇANDO"
9020 PRINT "A DISTANCIA OS SEUS M
ISSEIS SOBRE"
9025 PRINT "O INIMIGO, SO HABILM
ENTE VECERA,"
9030 PRINT "OS COMANDOS (A) (S)
(Z) (X) MOVEM A"
9035 PRINT "NAVE NAS 4 LINHAS DE
RUMO, O (L)"
9040 PRINT "LANCE O MISSEL QUE E
COMANDADO A"
9045 PRINT "DISTANCIA POR (J) (K)
(N) (M), O (Y)"
9050 PRINT "POR VEZES CONSEGUE L
ANÇAR A NAVE"
9055 PRINT "NO HIPER ESPACO, PAR
A SEU ALIVIO"
9060 PRINT "O (S) DIZ SIM A NO
VO JOGO,"
9063 PRINT "PRIMA **N/L** PARA
COMEÇAR"
9065 INPUT L$
9070 CLS
9079 RETURN

```

SUBMARINOS

Zx81 16 K

FERNANDO PRECES

Sacavém

```

0 REM PROGRAMA MODIFICADO POR
ALMEIDA PRECES EM 27/6/1982
3 REM "3"
5 LET T$="**PERSEGUIÇÃO AO SU
BMARINO**"
8 PRINT T$;"QUER INSTRUÇÃOE
S (S/N)?"
10 INPUT A$
15 IF CODE A$=56 THEN GOSUB 20
00
20 RAND
25 LET L=INT (10*RAND+1)
30 LET J=L
32 LET P=0
34 LET Q=0
36 LET R=0
40 DIM S$(L,2)
50 FOR I=1 TO L
60 LET S$(I,1)=CHR$ INT (19+RAND+
1)
70 LET S$(I,2)=CHR$ INT (19+RAN
D+1)
80 IF S$(I,1)=" " THEN GOTO 60
90 NEXT I
100 CLS

```



```

110 PRINT "INTRODUZA A VELOCIDA
DE EM "NDS" (15-LENTO, 30-RA
PIDO)";
120 INPUT T;
130 LET T=750/T;
140 IF T<100 THEN LET T=100;
150 IF T>5 THEN LET T=5;
160 CLS;
170 PRINT T;
180 FOR I=1 TO 19;
190 PRINT " ";
200 NEXT I;
220 PRINT "CARREGUE "S,6,7 E
8="";
230 PRINT AT 10,10;"TAB 21";
MOVIM."TAB 21";"CARGAS";"TAB 21";
"ABATES";
240 LET M=10;
250 LET N=M;
260 LET C=120;
280 LET F=1;
310 FOR I=1 TO (T+C);
315 NEXT I;
320 LET K$=INKEY$;
330 IF K$="5" OR K$="8" THEN GO
TO 400;
340 IF F THEN PRINT AT M,N," ";
350 LET M=M+K$="6" AND M(19)=1;
K$="7" AND M 1;
360 LET N=N+K$="3" AND N(19)=1;
K$="5" AND N 1;
370 PRINT AT M,N;" ";
380 LET C=C-1;
385 LET P=P+1;
387 PRINT AT 10,26,P;
390 GOTO 300;
400 IF F THEN GOTO 430;
405 FOR I=1 TO 7;
410 PRINT TAB 21;"TORPEADO";
415 NEXT I;
420 PRINT "CARREGUE "N,L"" F
ARA COMECAR";

```

```

422 INPUT L;
425 GOTO 550;
430 LET F=0;
440 LET C=30;
450 FOR I=1 TO J;
460 LET D=SOR (ABS (CODE S$(I)-
M)*2+ABS (CODE S$(I,2)-N)*2);
470 IF D<C THEN LET C=D;
480 LET L=L-NOT D;
490 IF NOT C THEN LET S$(I)="ZZ";
500 NEXT I;
502 LET Q=Q+1;
504 LET R=R+NOT C;
510 LET C=C+123*NOT C;
520 PRINT AT M,N,CHR$ (C+28);
522 PRINT AT 11,28,Q;
524 PRINT AT 12,28,P;
530 IF L=1 THEN GOTO 310;
540 PRINT AT 21,0;"ZONA LIMPA";
*** PARABENS *****
550 FOR I=1 TO 200;
555 NEXT I;
560 CLS;
570 GOTO 20;
2000 CLS;
2010 PRINT T;"10 SUBMARINOS I
NIMIGOS ESTAO SUBMEROS NA SU
A AREA-PATROILHA";"MANTENHA AS
TECLAS DO CURSOR";"CARREGADAS,SE
AS LARGAR, CARGAS DE PROFUNDIDA
DE SAO DISPARADAS, SE FOR ALVEJA
DO SERA DESTRUIDO, SENAO,O SONAR
DA-LHE A DISTANCIADO SUBMARINO
MAIS PROXIMO";"ATENCAO";"ES
TE SUBMARINO SABEA SUA POSICAO
E LANCA TORPEDOS,TEM QUE SE MOV
ER ANTES QUE ELE DISPARE E O DE
STRUA";"BOA SORTE";
2020 INPUT L;
2025 CLS;
2040 RETURN;

```

3D-O'S-X'S

NEWBRAIN

In. PERSONAL COMPUTER
Adapt. de ISABEL CRISTINA

X'S e O'S tridimensionais é um jogo para o micro-computador NEWBRAIN.

Joga-se numa grelha de 4x4x4. O utilizador joga com os X'S e o N.B. c/ os O'S.

A ideia consiste em criar uma linha com 4 cruces antes do N.B. fazer o mesmo com os O'S; as linhas vencedoras podem estar num único nível ou ao longo dos 4 níveis, na posição vertical, horizontal, ou em diagonal.

Para dar entrada de 1 movimento dá-se 1.º o nível, depois a coluna e finalmente a fila (todos entram c/ caracteres numéricos) (ver exemplo).

Depois de um lançamento ter sido feito, já não pode ser alterado; mas se a casa para onde jogar já estiver ocupada, o N.B. dir-lhe-à, e terá de repetir de novo o movimento.

O computador tem uma vantagem muito grande porque trabalha num logaritmo bastante eficiente. As jogadas feitas por ele encontram-se nas linhas 1600 — 1760. Cada lançamento possível é feito com 1 factor de vantagem, calculado na base de se prevenir

que o utilizador não complete nenhuma linha, formando ele o maior número possível; o cálculo é bastante eficiente e quase se chega a pensar que é impossível ganhar.

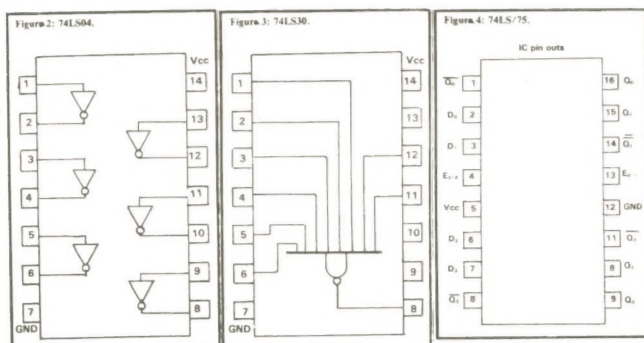
Para adaptação do programa a outros tipos de micro, apresenta-se seguidamente uma lista de comandos utilizados pelo N.B., neste programa e as respectivas funções.

Qualquer dos PUTS utilizados pode ser convertido em PRINT CHR\$(X) noutros micros (segundo o respectivo código). No N.B. só se utilizará PUT X.

- PUT 2 — Toda a linha lógica em que o cursor se encontra é apagada; as linhas abaixo sobem para preencher esse espaço.
- PUT 6 — A posição do cursor será sempre visível.
- PUT 7 — A posição do cursor deixa de ser visível.
- PUT 10 — Move o cursor para a linha física abaixo.
- PUT 21 — Quando seguido de "Get x, y", o cursor volta para a posição anterior.
- PUT 22 — Seguido de x, y, o cursor vai para coordenadas específicas.
- PUT 28 — Envia o cursor para a 1.ª posição da presente linha lógica.

Com o LATCH então ENABLED, a linha D7 deve passar ao nível lógico 1, que requer o valor 128 como o 2.º argumento do comando.

Àcerca do dispositivo 74LS/75, considera-se de bastante utilidade. É apenas de 4 bit LATCH, o que lhe permite ter 8 linhas DATA. São necessários muitas vezes 2 dispositivos mas o 7475 custa apenas 1/4 do preço de um 8 — bit LATCH como o 373 e tem outras características. As linhas são ENABLED em pares sendo fáceis de controlar; também para a linha INPUT há duas linhas OUTPUT c/uma linha invertida. Se traçarmos cada linha OUTPUT do Latch, apercebemo-nos que a linha invertida é usada para SINK-TAKE IN-CURRENT através de um Led e séries de resistências de 270 ohm. O Led funciona como um indicador, mostrando quando o RELAY está ligado. O OUTPUT LATCH, dando um nível lógico 1 (5V), é alimentado por uma resistência de 1K na base dum transistor BC-108. Comutará o transistor permitindo que a corrente passe através do relay para o colector do BC-108.



Uma resistência de 560 ohm em série com a bobina do relay reduz o consumo de corrente. Não esqueça o princípio básico de colocar um diodo através dos terminais da bobina, para assegurar o retorno da corrente quando se efectua o corte. Finalmente, observe que o cabo do "remote control" fica ligado entre os terminais de contacto do relay. Observe ainda que os terminais estão fechados quando o relay está "on" ou seja actuado.

Usando o comando do BASIC OUT 63,128 para colocar o relay em "ON" e usando o comando OUT 63,0 para colocar o relay "OFF", terá um método simples mas eficaz para comandar o seu gravador.

A ordem dos acontecimentos será:

- 1) LIGAR O CONTROLADOR NA PARTE DE TRÁS DO SPECTRUM
- 2) LIGAR A ALIMENTAÇÃO
- 3) OBSERVAR O "LED" — se estiver com luz, entrar com o comando OUT 63,0 através do teclado; ligar o cabo do "remote control" na ficha respectiva do gravador
- 4) LIGAR OS CABOS "EAR" / "MIC" DO GRAVADOR, AO SPECTRUM.

- 5) PREMIR A TECLA "PLAY" A FITA NÃO DEVE GIRAR!
- 6) QUANDO A FITA ESTIVER EM POSIÇÃO DE AVANÇAR, UTILIZAR O COMENDO DIRECTO.
OUT 63,128: LOAD""

Para os programas de sua autoria, fazer o SAVE com o LINE respectivo de forma a fazer com que o programa inicie automaticamente.

A primeira linha do seu programa, deverá ser por exemplo:

10 out 63,0 que irá originar a paragem do gravador.

Se é necessário colocar o gravador em funcionamento, terá então:

num. linha — OUT 63,128: MERGE "" seguido de
XXXX + 1 OUT 63,0: RUN 20 (ou GOTO 20).

OUTRA UTILIDADE: combinar o uso do controlador com a possibilidade de carregar outro ecran ou figura ... Por exemplo:

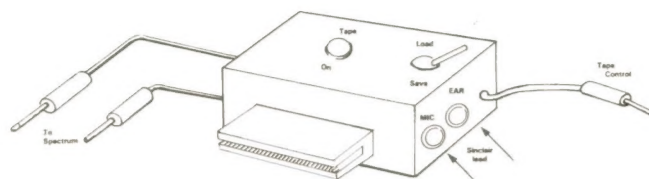
XXXX OUT 63,128: LOAD ""SCREEN
XXXX+1 OUT 63,0: GOTO XX

ATENÇÃO: XXXX +1 E NÚMERO DE LINHA A SEGUIR A NUM. XXXX.

RECORDE QUE DEVE COLOCAR SEMPRE comando OUT primeiro.

O CONTROLADOR DEVE SER COLOCADO NUMA CAIXA QUE PERMITA CORTAR A LIGAÇÃO DO "EAR" COM O GRAVADOR. Como é do conhecimento de muitos amigos nossos, quando se executa um SAVE (gravar), o cabo do "ear" deve estar desligado para evitar o "feedback".

O MESMO CONTROLADOR PODE SER USADO COM O ZX81 OU TMS 1000. APENAS DEVE SER USADA UMA PEQUENA ROTINA EM CÓDIGO MÁQUINA PARA O COMANDO "OUT" ... QUEM O DESEJAR DEVE PORTANTO, ESCREVER-NOS.



NOVOS PROGRAMAS

SPECTRUM

- **STRIKE ATTACK** (48 K) — És um piloto de um "Phantom" numa base aérea da NATO. A tua missão é destruíres pontos-chaves no território inimigo.
- **WHEELIE** (48 K) — O "Rodinhas" conduz a sua motorizada no circuito de Motocross.
- **MR. WIMPY** (48 K) — Ele propõe-se fazer os melhores Hamburguers da cidade. Para combater os personagens que o importunam, terá que fugir ou atirar-lhes com pimenta.
- **DAMAS** (48 K) — Prática do tradicional jogo de damas, com a diferença de a dama só se poder deslocar de casa a casa (e não em toda a fila).
- **HORACE AND THE SPIDERS** (48 K) — Inicialmente, Horace tem que trepar os montes, fugindo das aranhas. Para atravessar fendas, usa as teias que estas fazem. Mas elas não perdoam! Uma demora de Horace pode significar a sua morte.
- **GOLF** (48 K) — Podes escolher entre 15 clubes e jogar normalmente.
- **JACKPOT** (48 K) — Simulação de uma máquina de fruta, com explicação completa das suas regras: HOLD, NUDGE, GAMBLE e FEATURE BOX.
- **EVEREST ASCENT** (48 K) — Escalar a montanha mais alta do mundo não é tarefa fácil! Cuidado com as avalanches, o gelo, o relevo, as cavidades...
- **SUPER SPY** (48 K) — Localiza a ilha secreta do misterioso Dr. Death (cuidado com o seu criado!). Terás que atravessar continentes cheios de puzzles e labirintos complicados.
- **TRANSYLVANIA TOWER** (48 K) — O objectivo é chegar ao topo da torre, matar o conde Kreepie e apanhar o tesouro. O percurso é complicado: São 5 andares, com 100 quartos cada e com características diferentes. Atenção aos vampiros e objectos!
- **CHUCKIE EGG** (48 K) — Protege bem os teus ovos e o milho antes que tos destruam e devorem.
- **PAINTER** (48 K) — Vais percorrer um labirinto, transportando baldes que deverás encher com tinta. Há um rolo que te persegue e, por cada balde que consigas encher, o número de rolos aumenta também...
- **KONG** (48 K) — O gorila capturou a donzela. Passarás por barreiras bem difíceis (bolas de fogo, etc.) para a libertar.
- **COSMIC RAIDERS** (16 K) — És um piloto de uma nave e vais defender a Terra do ataque de seres estranhos que já instalaram uma base no teu planeta.
- **JETPAC** (16 K) — O astronauta pretende recolher o máximo possível de minerais valiosos em vários planetas. E, quanto mais andar, mais os habitantes tentam detê-lo. Além disso, o combustível pode acabar.
- **JUMPING JACK** (16 K) — Jack tem que saltar do fundo do écran para plataformas superiores; Mas só o consegue se houver aberturas. Quanto mais subir, mais altos são os níveis das plataformas. Se cair num nível inferior, perde a vida.

(Preço/programa — 400\$00)
(excepto STRIKE ATTACK)

DESCONTO DE 20% PARA SÓCIOS DO CLUBE

NOVO LIVRO

- **MICRODRIVE AND INTERFACE 1 MANUAL** (em Português), Sinclair Research Limited, England, 1983 — Ed. e Trad. de LANDRY.

(Preço — 200\$00)



CLUBE Z₈₀

INSCRIÇÃO COMO ASSOCIADO

O **CLUBE Z₈₀** está aberto a todos os utilizadores de microcomputadores.

A intenção de associar os entusiastas das micro-máquinas, é exclusivamente a de permitir:

- 1 — PUBLICAÇÃO DE UM JORNAL MENSAL, onde sejam publicados programas de uso geral ou específico como no caso da educação.
- 2 — PROMOVER TROCAS DE PROGRAMAS, e trocas de experiências; tanto no caso do Software (programação), como no caso do Hardware (electrónica).
- 3 — PROMOVER DESCONTOS NA AQUISIÇÃO DE PROGRAMAS.
- 4 — LANÇAR CURSOS DE PROGRAMAÇÃO EM BASIC — PASCAL OU OUTRAS LINGUAGENS E DIVULGAR O USO DE LINGUAGEM MÁQUINA.

NOME _____

IDADE _____ COMPUTADOR TIPO _____

PROFISSÃO _____

ENDEREÇO _____

TELEF. _____

ASSINATURA ANUAL — Esc. 1 500\$00 ☐

ASSINATURA SEMESTRAL — Esc. 750\$00 ☐

CHEQUE OU VALE DO CORREIO

N.º _____

BANCO _____

DATA ____/____/____

JÁ SÓCIO ☐

NOVO SÓCIO ☐ → A partir do mês de _____ (inclusive)